

# Review of some classical gravitational superenergy tensors using computational techniques

A. Balfagón\* and X. Jaén.†

February 7, 2008

## Abstract

We use computational algorithms recently developed by us to study completely four index divergence free quadratic in Riemann tensor polynomials in GR. Some results are new and some other reproduce and/or correct known ones. The algorithms are part of a *Mathematica* package called *Tools of Tensor Calculus (TTC)* [web address: <http://baldufa.upc.es/ttc>]

## 1 Introduction

Recently the interest of tensors containing the interaction fields, and in particular the Riemann tensor, has been increased. The so-called superenergy tensors are good examples, having some particular properties [1]. This interest is not new. The tensors of Bel-Robinson[2], Bel[3] and Sachs[4] among others are some examples which have been studied in the past and having zero divergence in certain conditions. CD Collinson [5] has made a much deeper study giving all four index quadratic in Riemann and divergence free tensors in four dimension for any space-time. Nevertheless this kind of calculations are very hard to do by hand and some errors could arise. Having this in mind, recently, we have developed some computational algorithms able to handle index tensor polynomial expressions. The paradigm of these polynomials are those built with Riemann tensor as in the example

$$\dots \frac{1}{2} R_{ij}{}^{mn} R_{mn}{}^{kl}{}_{;l} + 3 R_{im}{}^{km} R_{j;l}{}^l \dots$$

The implementation of the algorithms are part of a package called *TTC* [6] (*Tools of Tensor Calculus*). These algorithms allows us to perform a complete

---

\*Institut Químic de Sarrià, Laboratori de Física Matemàtica, Societat Catalana de Física (I.E.C.) Universitat Ramon Llull e-mail: [abalf@iqs.url.es](mailto:abalf@iqs.url.es)

†Universitat Politècnica de Catalunya, Laboratori de Física Matemàtica, Societat Catalana de Física (I.E.C.) e-mail: [jaen@baldufa.upc.es](mailto:jaen@baldufa.upc.es)

simplification of this kind of expressions. The simplification algorithms do not find the most beautiful expression but, besides zero result, a canonical one.

Other authors who have worked on algorithmic simplification of index tensor polynomials are Ilyin and Kryuchov[7], Portugal[8], Fulling et al.[9]. Parker and Christensen[11] do not fall on this list because although they have a big package on tensor calculus, they don't work algorithmically in this topic but with libraries of rules. In a recent article[10] we have solved the part of the simplification problem related to dummy indices and monotermin symmetry properties. Cyclic and in general multitermin symmetries properties were not considered. Also dimensional properties were left aside for further works, due to its complexity. Actually [12, 13] we can handle multitermin and dimensional properties with the advantage that the algorithm includes the possibility to store a compacted version of the generated rules in order to increase functionality. This means that, although a first computation over a polynomial can be really slow, others on the same or on other sessions are faster than the first one. This allows us to *play* easily with some results by permuting indices or making some other operations.

There are some trust problems when using programs which do automatic computations, specially when no exact coincidences arise when compared with older papers. All we can say is that our algorithm has been tested, besides in the present paper, on a large set of known Riemann properties ( among others all those included in [11]) and calculations related to the Gauss-Bonnet invariant[14]. On the other hand the fact that the implementation of the algorithm is part of a package, *TTC*, ( from now on we will use the name *TTC* to call both, the simplification algorithm and/or the package) it allows us to test the results on explicit space-times. We have tested the calculations on Vaydia metric confirming our results.

The aim of this paper is:

- To show the *TTC* capabilities on a real calculus. This can be of interest to researchers working on algebraic computation and/or working on superenergy tensors or on any other field where these techniques can be applied.
- To review some known results and correct some others. Specially we have corrected a minor part of the results of C.D. Collinson and some sign on the original Sachs paper [4].
- To prove some theorems about existence and uniqueness of superenergy tensors.

By superenergy tensors we understand, in this paper, four index quadratic in Riemann tensor having some chosen properties. Following the work of C.D. Collinson we will find divergence free tensors and from these we will analyze the index symmetry properties. We leave aside other properties such as the dominant super-energy property (DSEP) and others explained in reference[1]

The plan is as follows. In section 2 we briefly explain what *TTC* does when simplifying index tensor polynomials in general. Also we describe some *TTC*

functions usually used in operations over this kind of objects . In section 3 we define exactly what kind of tensors we will find and which are the steps to reach our main result, that is to find all four index quadratic in Riemann and divergence free tensors in any dimension and in any space-time. Next we specialize these results for four dimensions and for vacuum space-times. We analyze the tensors found with respect to index symmetry properties, specially we give existence and uniqueness theorems for totally symmetric tensors.

Finally Appendix A is devoted to show a sample *TTC* session where the C.D.Collinson calculations are reproduced using the essential *TTC* functions.

## 2 Computational simplification of index tensor polynomials

As stated above, by simplification we mean to find a canonical, therefore unique, index tensor polynomial for any given monomial. In this process the following must be taken into account:

- Dummy indices: the output expression must be unique with respect to the freedom of renaming dummy indices.
- Index symmetry properties of each singular tensor appearing in the monomial: since in this paper we work with Riemann tensor  $R$  the following properties, and those which arise when differentiating and contracting, must be taken into account

$$\begin{aligned}
R_{ijkl} &= -R_{jikl} = -R_{ijlk} = R_{klij} \\
R_{ijkl} + R_{iljk} + R_{iklj} &= 0 \\
R_{ijkl;m} + R_{ijmk;l} + R_{ijlm;k} &= 0 \\
R_{ijkl;\underbrace{m,\dots,n}_k} &= a R_{ijkl;\underbrace{n,\dots,m}_k} + (\text{Riemann terms})_{\underbrace{\dots}_{k-2}}
\end{aligned} \tag{1}$$

where the last one refers to Ricci commuting relations. At this point it is important to set up the convention for Ricci commuting relations and the definition of Ricci tensor used by *TTC* which follows [15].

If  $v$  is a vector field then

$$\begin{aligned}
v_{i;j;k} - v_{i;k;j} &= v_m R^m_{i j k} \\
R_{ij} &= R^m_{i m j}
\end{aligned}$$

It is important to note the change of sign of the Ricci tensor with respect to that used in Sachs[4], Collinson[5] and Robinson[17] original papers

- Properties due to the finite dimension of space. This kind of properties arise due to the possibility of antisymmetrization over  $n+1$  indices appearing in the monomial,  $n$  being the dimension of space. In the next example

$$n = 2$$

$$\dots + A_{ij} B_k + \dots \Rightarrow A_{[ij} B_{k]} = 0$$

In the following subsection we briefly explain how algorithms work. For details we refer to [12, 13]

## 2.1 Dummy index problem

Actually *TTC* uses a codification of each monomial, which is an improved version of [10], following the criteria:

- 1) Free indices are renamed: `IndexF[1]`, `IndexF[2]`, ... all of them in upper position (contravariant)
- 2) Dummy indices are renamed `IndexD[1]`, `IndexD[2]`, ... all of them in upper position (contravariant)
- 3) We find the permutation of free and dummy indices in order to find the most lexicographically ordered monomial expression. Note that at this stage the codification is independent of the order of free indices. We call this *globalcodification*

Finally we take the invers of permutation performed on free indices over the *globalcodification* of the monomial. The final result is called *codification* of the original monomial.

All rules found through a calculation are stored in *codification* version but all monomials with the same *globalcodification* version can use a permutation version of the stored rules, so the work of finding rules is essentially independent of the order of free indices.

## 2.2 Index tensor properties

First the user must declare each tensor entering the monomial, together with their symetries. Some properties are automatically applied as Ricci commuting relations.

Key functions: `InputTensor`, `InputSymmetries`

*TTC* acts on each monomial applying in all possible ways the individual properties of each tensor over the starting monomial and over each new resulting monomial. When this loop ends (no new monomials are created) *TTC* solves the resulting system of equations and applies the rules over the original monomial. Subsequent monomials can use the same or a permuted version of these rules if the *globalcodification* version is the same as one of the monomials in the rules.

Key functions: `Index[gname, SimplifyAllIndex[2]]`

## 2.3 Dimensional properties

As stated in the introduction, dimensional propeties arise due to the possibility of antisymmetrizing  $n + 1$  indices of the starting monomial and equating the result to zero,  $n$  being the space dimension. Nevertheless we must be sure to take all possible sets of  $n + 1$  indices in the monomial, even if they are dummy or

*hidden* ( as in the curvature  $R$ ). The algorithm takes the monomial and writes it in a metric expanded version. The possible sets of  $n + 1$  indices are in the metric part of the monomial. Let us see some examples in four dimensions.

1) In the first example there are no dimensional properties:

$$R \rightarrow R_{mnpq} g^{mp} g^{nq}$$

2) Only one set of 5 indices can be effectively antisymmetrized:

$$R R^{ab} \rightarrow R_{mrns} R_{opqr} g^{mn} g^{ar} g^{bs} g^{oq} g^{pr}$$

3) The last example is the case handled in the present paper. Several sets of 5 indices can be effectively antisymmetrized. Note that for  $n \geq 6$  there are no dimensional properties.

$$R R^{abcd} \rightarrow R_{mnop} R_{rstv} g^{mo} g^{np} g^{ar} g^{bs} g^{ct} g^{dv}$$

*TTC* finds all these properties and simplify them using nondimensional ones. Then it solves the resulting system of equations and applies the resulting rules when necessary. Although the procedure described is correct actually the algorithm has been improved in order to optimize the number of sets of indices to be antisymmetrized [13]

Key functions: `Index[gname,SimplifyAllIndex[3]]`

### 3 Collinson like calculus

C.D Collinson found *by hand* 10 independent divergence free tensors having four indices and quadratic in Riemann. Here we use *TTC* to make the computerized version of his calculations, but the plan is close to the Collinson paper.

First we must generate the starting basis

$$T^{abcd} = A^{abcdefg hklmn} R_{efgh} R_{klmn} + B^{abcdklmnpq} R_{klmn;pq}$$

where  $A^{abcdefg hklmn}$  and  $B^{abcdklmnpq}$  are polynomic tensors built with the metric  $g^{ab}$  using the free indices plus the contracted ones in all possibles ways, symbollically

$$A^{abcdefg hklmn} = \sum_{\sigma} \alpha_{\sigma} \sigma (g^{ab} g^{cd} g^{ef} g^{gh} g^{kl} g^{mn})$$

$$B^{abcdklmnpq} = \sum_{\sigma} \beta_{\sigma} \sigma (g^{ab} g^{cd} g^{kl} g^{mn} g^{pq})$$

$\sigma$  being the permutations over metric indices and  $\alpha_{\sigma}$  and  $\beta_{\sigma}$  constant parameters.

*TTC* performs this calculation simplifying the result using nondimensional properties and dimensional ones. In the second case we obtain, as in Collinson's paper, 63 independent monomials. This polynomial is what we call the starting basis  $T$ , which has 63 independents coefficients (including the global one). In

doing this  $TTC$  finds 3 independent dimensional properties which includes those found by H.A.Buchdahl[16].

Second we take the divergence of  $T$

$$T^{abcd}{}_{;a}$$

$TTC$  calculates this and simplifies it by using nondimensional and dimensional properties. After this we must solve the coefficients of the equation

$$T^{abcd}{}_{;a} = 0 \quad (2)$$

so we find the family of polynomials  $T$  fulfilling (2). This computation has been done in a generic space-time and in vacuum. The results have been analyzed under a few relevant index symmetry properties. From the computations we have:

### 3.1 Without-dimension properties

**Theorem 1** *In a generic space-time and not using dimensional properties:*

1) *There exist 14 independent quadratic in Riemann four index divergence free tensors.*

2) *There are no totally symmetric tensors fulfilling 1)*

3) *The complete family of tensors  $T^{abcd}$  fulfilling 1) totally symmetric in  $(bcd)$  is*

$$\begin{aligned} T^{abcd} &= a_S T_S^{abcd} + a_R T_R^{abcd}, \\ T_S^{abcd} &= Q^{a(bcd)}, \\ Q^{abcd} &= -\frac{1}{3} g^{ac} R^d{}_i R^{ib} + 2 R^{ab;cd} - \frac{4}{3} R^{bd;ac} + \\ &\quad \frac{4}{3} g^{ac} R^{bd;i}{}_i - 2 g^{ac} R^b{}_i{}^{;d}{}_i + \frac{4}{3} R^b{}_i R^{acid} + \\ &\quad 2 R^{aibj} R^c{}_i{}^d{}_j - \frac{1}{2} g^{ac} R_{ij}{}^d{}_k R^{ijk} \\ T_R^{abcd} &= X^{ab} g^{cd}, \\ X^{ab} &= K U^{ab} + L V^{ab} - \frac{1}{4} W^{ab} \\ U^{ab} &= G^{ab;s}{}_s - 2 G^{sb;a}{}_s + 2 G^a{}_p R^{pb} - \frac{1}{2} G^a{}_{pq} R^{pq} g^{ab} \\ V^{ab} &= R^{;ab}{}_s - R^{;s}{}_s g^{ab} - R S^{ab} \\ W^{ab} &= G^{apqr} R^b{}_{pqr} - \frac{1}{4} g^{ab} G^{mpqr} R_{mpqr} \\ G^{ab}{}_{cd} &= R^{ab}{}_{cd} - 4 g^{[a}{}_c S^{b]}{}_d \\ S^{ab} &= R^{ab} - \frac{1}{4} g^{ab} R \end{aligned}$$

where  $a_S, a_R, K, L$  are four independent constant parameters.

$T_S$  is the tensor found by Sachs[4] with an opposite sign on the term  $\frac{4}{3} R^b{}_i R^{acid}$  with respect to his original definition.  $T_R$  is the family defined by Robinson[17].

**Theorem 2** *In a vacuum  $R_{ij} = 0$  space-time and not using dimensional properties:*

1) There exist 6 independent quadratic in Riemann four index divergence free tensors.

2) There is only one tensor  $T^{abcd}$  fulfilling 1) and totally symmetric:

$$\begin{aligned} T^{abcd} &= M^{(abcd)} + 16 P^{a(bcd)} - 4 Q^{(abc)d} - 4 Q^{(a|d|bc)}; \\ M^{abcd} &= g^{ab} g^{cd} R_{mnop} R^{mnop}; \\ P^{abcd} &= R_m^a{}^n{}_b R^{mcnd}; \\ Q^{abcd} &= g^{ab} R_{mno}^c R^{mnod} \end{aligned}$$

### 3.2 Using dimensional properties

**Theorem 3 (Collinson corrected)** *In a generic four dimensional space-time:*

1) There exist 9 independent quadratic in Riemann four index divergence free tensors.

Using the same notation of Collinson [5] the Collinson family is

$$T^{abcd} = a_1 T_1^{abcd} + a_2 T_2^{abcd} + a_3 T_3^{abcd} + a_4 T_4^{abcd} + a_5 T_5^{abcd} + a_6 T_6^{abcd} + a_8 T_8^{abcd} + a_9 T_9^{abcd} + a_{10} T_{10}^{abcd}$$

where  $a_i$  are arbitrary numerical coefficients.

The tensors  $T_1$  to  $T_9$  (without  $T_7$ ) can be written in terms of  $T_{10}$

$$\begin{aligned} T_1^{abcd} &= T_2^{acbd} = T_3^{adcb} = g^{cd} T_{10}^{a i b}{}_{i} \\ T_6^{abcd} &= T_5^{acbd} = T_4^{adcb} = g^{cd} T_{10}^{a b i}{}_{i} - \frac{1}{2} g^{cd} T_{10}^{a i i}{}_{i}{}^b \\ 3 T_8^{abcd} &= T_{10}^{acbd} - T_{10}^{abcd} \\ T_9^{abcd} &= T_{10}^{adcb} - 2 T_{10}^{abcd} \end{aligned}$$

and  $T_{10}$  is defined through  $T_{10}^{abcd} = 6 Q_1^{abcd} + Q_2^{ab(cd)} + Q_3^{ab(cd)}$  being

$$\begin{aligned} Q_1^{abcd} &= R^c{}_i R^{bdai} + R^b{}_i R^{daci} - R^{aibc}{}_{;i}{}^d + R^{acdi}{}_{;i}{}^b + g^{ab} R^{dicj}{}_{;ij} \\ Q_2^{abcd} &= 4 R^{aibd}{}_{;i}{}^c - 6 g^{ad} R^{ib;c}{}_{;i} - 6 g^{ad} R^{icbj}{}_{;ij} + 6 R^{ab;cd} + 6 R^{abci}{}_{;i}{}^d \\ Q_3^{abcd} &= 8 g^{ac} R_{ij}^d{}_k R^{jkbi} - 8 R_{ij}^{cd}{}_j R^{biaj} + 8 R_i^d R^{bica} - 8 R_i^{bd}{}_j R^{ciaj} + \\ &\quad 2 R_i^d R^{baic} - 2 R_i^{db}{}_j R^{ciaj} - 3 g^{ab} R_i^d R^{ci} + 3 g^{ab} R_{ij}^d{}_k R^{jkci} \end{aligned}$$

2) There are no totally symmetric tensors fulfilling 1)

We note that the Collinson tensor  $T_7$  does not appear, which is identically zero, since  $T_{10}$  is in fact symmetric with respect to the last two indices (this is not mentioned in Collinson's paper).

**Theorem 4** *In a vacuum  $R_{ij} = 0$  four dimensional space-time:*

1) There exists 1 quadratic in Riemann four index divergence free tensor.

2) The tensor found in 1) is totally symmetric and is the Bel-Robinson tensor [2].

## 4 Concluding remarks

Trough four theorems we have reviewed and improved some results on classical gravitational superenergy tensors. The results have been reached using computational algorithms which do automatic calculations. As we have commented in the introduction some trust problems could arise.

Nevertheless we think that trust like problems are not exclusively related to computational techniques. How can we be sure that *all* calculation of C.D. Collinson, or anyone, are correct ?. Collinson is a good pioneer on large calculations on the topic but to validate his results we hope some other should obtain the same results using independent calculations. We also hope that any other program will be able to do our calculations with an independent code and/or algorithms...

Assumming that these computational techniques are validated and limiting our scope to superenergy tensors we think that they are a very good complement to research based on defining such tensors having some choosen properties. We can analyze the uniqueness of these definitions with respect to the desired properties. But we can also analize more complex problems, specially those related to the existence of conserved quantities. These studies can be done on a generic space- time and using Einstein field equations. We hope that in the near future we will be able to handle the case of cosmological vacuum and scalar field interaction.

## Acknowledgments

We want to kindly thank J.M.M.Senovilla to direct our work on superenergy tensors. X.J. would like to thank the Comisión Asesora de Investigación Científica y Técnica for partial financial support, under Contract No. PB96-0384. The authors would also like to thank the Laboratori de Física Matemàtica at the Societat Catalana de Física for partial financial support.

## Appendix: The *TTC* master session

This appendix is devoted to show a sample of *TTC* session so that somebody else can reproduce the calculations using *TTC*. In this session we find the result of Theorem(3). The inputs are indicated by `In[]:=`. The outputs have no special sign. The comments are enclosed by `(* comments *)`.

```
(* first we load the program ttc.m. we load also eimes.m
where some private tools are defined. *)
In[]:
<<ttc.m
<<eimes.m
-----
|Tools of Tensor Calculus 4.1.0 |
| A.Balfagon, P.Castellvi and X.Jaen |
| \protect\vrule width0pt\protect\href{http://baldufa.upc.es/ttc}{http://baldufa.upc.es/ttc} |
| e-mail: ttc@baldufa.upc.es |
```



```

|      version: september,XX,1999      |
|-----|
|      Session started on              |
|      September, XX, 1999             |
|      at 16 h 8 min 52 s              |
|-----|
CPU Time=0. s Memory in use=2.434 Mb
|-----|
|      Eines Personals 1.0             |
|      X.Jaen & A.Balfagon            |
|      version: juny, 23, 1999         |
|-----|

(* Here we declare the name of the coordinates and dimension to be used when needed. We also introduce a
metric and a Riemann tensor.
Finally we introduce the indices we want to see in outputs *)

In[]:=(
InputCoordinates[cx4,4];
ScalarBasis=cx4;ScalarBasisQ=False;
InputSMetric[gn,cx4,"g",gl];
InputSRiemann[gn,cx4,"R",Rie,Ric,R];
InputIndex[{i,j,k,l,m,n,o,p,q}]);

(*This is usefull if we want to store in "filename" all rules
generated by TTC. If you use it remember end save using
EndSimplifyAllIndexSave*)

(*IniSimplifyAllIndexSave[filename]*)

(* generator is where the kind of tensors we study are defined*)

In[]:=generator=
Rie[a,b,c,d] Rie[-a,-b,e,f]+Rie[a,c,d,e,.-a,.;f]//Index[gn]

      i j k l      k l      m n i j      m i j k      .;l
0      + R      R      + R
      m n      .;m
(* IndexTensorBasis takes the generator and uses some easy tricks
to build the starting basis, simplifying them using
nondimensional and dimensional properties. It uses A[number]
as parameters. The result is stored as base*)

In[]:=generator//Index[gn,IndexTensorBasis[3,A,base,CoRiemannRules[gn,cx4]]];

(* see *)
(*EndSimplifyAllIndexSave*)
(* we can see the starting basis)
In[]:=base

      i j k l      .;k .;l      i j      .;j .;l      i k
0      + A[1] R      g      + A[2] R      g      +.....
      m n      .;m
> A[62] g      g      R      m n o p      R      i j      k l      R      m n o p
      m n o p      m n o p      m n o p

(* Here we define A[1].. A[63] as constants parameters*)
In[]:=SetConstantCoeff[A,cx4,63];

(* Here we compute the divergence of base and simplify up to
dimensional properties assigning the result to dbase *)

In[]:=dbase=base[.,-i]//Index[gn,SimplifyAllIndex[3]];

(* Here we find the parameters which solves dbase=0 assigning
the result to rules*)

```

```

In[]:=rules=SolveCoeff[dbase,A][[1]];

(*Here we apply rules to the starting basis base assigning
the result to baseXA*)

In[]:=baseXA=base/. rules;

(* baseXA fulfill the condition imposed...*)
In[]:=baseXA[.;-i]//Index[gn,SimplifyAllIndex[3]]
0

(* Here we find t1xa ...t9xa which are the nine independent tensors
we have found corresponding to the nine independent coefficients A[number]
appearing in baseXA*)

In[]:=t1xa=baseXA/.
{A[6]->1,A[9]->0,A[12]->0,A[58]->0,A[59]->0,
A[60]->0,A[61]->0,A[62]->0,A[63]->0};

In[]:=t2xa=baseXA/.
{A[6]->0,A[9]->1,A[12]->0,A[58]->0,A[59]->0,
A[60]->0,A[61]->0,A[62]->0,A[63]->0};

In[]:=t3xa=baseXA/.
{A[6]->0,A[9]->0,A[12]->1,A[58]->0,A[59]->0,
A[60]->0,A[61]->0,A[62]->0,A[63]->0};

(* also for tx4,tx5,tx6,tx7,tx8*)

In[]:=t9xa=baseXA/.
{A[6]->0,A[9]->0,A[12]->0,A[58]->0,A[59]->0,
A[60]->0,A[61]->0,A[62]->0,A[63]->1};

(* we can see some of them*)
In[]:=t4xa

```

$$\begin{aligned}
& 0 \quad i \quad j \quad k \quad l \quad + \quad 2 \quad i \quad l \quad j \quad k \quad - \quad 2 \quad i \quad k \quad j \quad l \quad + \quad 2 \quad j \quad l \quad i \quad k \quad - \\
& 2 \quad j \quad k \quad i \quad l \quad - \quad 2 \quad i \quad l \quad j \quad k \quad + \quad 2 \quad i \quad k \quad j \quad l \quad - \quad 2 \quad j \quad l \quad i \quad k \quad + \\
& 4 \quad i \quad l \quad j \quad m \quad k \quad + \quad 2 \quad j \quad k \quad i \quad m \quad l \quad - \quad 4 \quad i \quad k \quad j \quad m \quad l \quad - \\
& 2 \quad i \quad l \quad j \quad k \quad R_{m \quad n} \quad + \quad 2 \quad i \quad k \quad j \quad l \quad R_{m \quad n} \quad - \quad 2 \quad i \quad k \quad j \quad l \quad + \\
& 2 \quad i \quad l \quad j \quad k \quad + \quad 2 \quad j \quad k \quad i \quad l \quad - \quad 2 \quad i \quad l \quad j \quad k \quad + \\
& 2 \quad j \quad l \quad i \quad k \quad + \quad 2 \quad i \quad k \quad j \quad l \quad - \quad 2 \quad j \quad l \quad i \quad k \quad + \\
& 2 \quad k \quad m \quad i \quad j \quad l \quad + \quad 2 \quad j \quad m \quad i \quad k \quad l \quad + \quad 2 \quad i \quad l \quad m \quad j \quad n \quad k \quad - \\
& 2 \quad i \quad k \quad m \quad j \quad n \quad l \quad - \quad i \quad l \quad m \quad n \quad j \quad k \quad + \quad i \quad k \quad m \quad n \quad j \quad l \quad +
\end{aligned}$$

$$R_{m n}^{i j} R_{m n k l}^{m n k l} - g^{j l} R_{m n o}^{i m n o k} + g^{j k} R_{m n o}^{i m n o l}$$

```
(* now we suppose that we have find baseC which is
the baseXA expressed ussing the basis found by Collinson.
We use B[1]...(B[7])...B[10] to name the parameters ( without B[7])
accordingly to Collinson notation. We take baseC and use
IndexUpdate to define T in such a manner that we can use T
as a tensor in their own right. This will permit us
to study symmetry index*)
```

```
In[]:=IndexUpdate[T,":=",baseC]
```

```
(*Here we study the symmetry between the first two indices,
simplifying them up to dimensional properties and finding
the condition for the parameters B[n]*)
```

```
In[]:=T[a,b,c,d]-T[b,a,c,d]//Index[gn,SimplifyAllIndex[3]];
```

```
In[]:=rules=SolveCoeff[%,B][[1]]
```

```
{B[4] -> 0, B[5] -> 0, B[2] -> 0, B[3] -> 0, B[8] -> 0, B[9] -> 0, B[10] -> 0}
```

```
(* Only B[1] and B[6] are independent.
It is in this way how we have analized
symmetry properties in this paper*)
```

## References

- [1] J.M.M.Senovilla gr-qc/9906087
- [2] Bel, L., *C. R. cad. Sci.*, Paris, **247** ,1094, 1958
- [3] Bel, L., *C. R. cad. Sci.*, Paris, **248** ,1297, 1959
- [4] R. K. Sachs, *Zeitschrift fur Physik*, **157**, 462-477,1960
- [5] C. D. Collinson, *Proc. Camb. Philo. Soc.*, **58**, 346,1962
- [6] A.Balfagón, X.Jaén,P.Castellví, *Tools of Tensor Calculus (TTC)*, *Mathe-*  
*matica* package, web adress <http://baldufa.upc.es/ttc>
- [7] V.A.Ilyin and A.P.Kryukov *Computers in Physics Communications* **96**,  
36.(1996)
- [8] R.Portugal To be published on Journal of Physics A
- [9] S.A. Fulling, R.C. King, B.G. Wybourne and C.J.Cummins ,  
*Class.Quantum Grav.* **9**] 1151-1197 (1992)
- [10] A. Balfagón, X. Jaén, *Computers in Physics*, **12**, 3 , 286,1998
- [11] L.Parker and S.M.Christensen, *Mathtensor: A System for Doing Tensor*  
*Analysis by Computer* , Addison-Wesley Publishing Company (1994)

- [12] A.Balfagón, X.Jaén preprint
- [13] A.Balfagón, X.Jaén preprint
- [14] S.M.Christensen, *The Mathematica Journal*, **1**, 1 (1990)
- [15] C. W. Misner, K. S. Thorne, J. A. Wheeler, *Gravitation*
- [16] Buchdahl, H.A., *J. London Math. Soc.*,**26**,150 (1951)
- [17] I.Robinson, *Class. Quantum Grav.* **14** (1997) A331-A333